

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/221610488>

Application of Software Engineering Fundamentals: A Hands on Experience.

Conference Paper · January 2005

Source: DBLP

CITATIONS

7

READS

511

3 authors, including:



Cihan Varol

Sam Houston State University

74 PUBLICATIONS 402 CITATIONS

SEE PROFILE

Application of Software Engineering Fundamentals: A Hands on Experience

Cihan Varol¹, Coskun Bayrak¹, and Robert Ludwig²

¹University of Arkansas at Little Rock, Computer Science Department

2801 S. University Avenue, Little Rock, AR 72204

E-mail: cxvarol | cxbayrak @ualr.edu

² Acxiom Corporation, Inc.

1 Information Way

Little Rock, AR 72202

robert.ludwig@acxiom.com

Abstract— Software engineering industry is improving, hence, the expectations from the employee is always increasing. In today's software engineering notion not only expected from employee to be successful in technical background, but also the candidate needs to handle non-technical issues arises from complex project. Requirements gathering, working in a team, professionalism, handling pressure, and scaling themselves to deadlines are some of the non-technical issues that employee faces. Therefore, it is obvious that, teaching software engineering can not be covered only theoretically from the text book. Students need to experience a client-sponsored software engineering project to understand all fundamentals of a software development. This article provides detail information about the methods that were used and experiences gained from a real-world software engineering project in a class environment.

Index Terms— Client-Sponsored Project, Focus Group, Software Engineering, Software Management, Software Requirements

I. INTRODUCTION

The first computer, which was built by Pennsylvania University, weighed 40 tons and it bestrewed over 1600 sq-ft place to engage for army purposes [1]. Appearance of the first computer engender to the born of first generation of software engineering industry. Since the assessment of first computers around 1940's, the technology of microchips and computer devices has been improved incredibly rapid and old-fashioned computers replaced with today's high-tech, fully equipped small intelligent affluences. Not only had the hardware pieces, but also software engineering companies have been renewed them to produce the efficient, user friendly programs to their costumers.

Software development has ballooned into a multibillion dollar industry due to the demand of software in government and private sectors. Necessity of the software development and expectation of the quality will be always lofty while sectors move more into information age. Companies desire to hire skilful, neat, diligent and real-life project experienced

candidates to join their team. However, although most of the software engineers take courses in software engineering and programming from their educational institution, generally the employees face with real-life software engineering tasks while they are having a job at software developer company. It populates muddle between the concepts of theoretical and practical approaches in software engineering at the educational institutions.

Many of the universities are offering software engineering as a course in their Computer Science curriculum. Globally, the demand of software engineers is far exceeding the supply of software engineer graduates. Most often, this gap is filled by computer science graduates. Because software developer companies prefer experienced candidates, it is very significant to give opportunities to the students to practice software engineering concepts and principals through a real-life software development project.

Different approaches have been applied for providing real-world experience. In one of them the software engineering and database classes combined together and built an assisting to a local lactation consultant who was dissatisfied with the software product [2]. Another one divides the students into two parts, one group is in the client-side of the development process, and the other group is responsible for the development [3]. In common, almost all strategies plead to using real-world projects in software engineering classes.

At University of Arkansas at Little Rock (UALR), software engineering courses are offered to both undergraduate and graduate students by computer science department. Generally the graduate level software engineering course is held in two semesters of the year. This course is a 3 credit course and 3 hours of lecture per week are delivered. Starting from the requirements engineering to software testing techniques, all design and development steps taught by the instructor. Relevant homework assignments are carefully selected and given to students regarding the related topics, so that the fundamental principles are clearly understood and applied when appropriate throughout the project development phases.

The real difference in SE teaching at UALR is the demanding project component. At UALR, the software

engineering students work on client-sponsored project or projects based on the number of students in the class. Students get most of their final grade based on their efforts and final product from the project. Since there was a team of 5 students, all students were assigned to only one project with a project leader and a project advisor. In this paper, the class session times, their influence to the process of the development of the project and the effectiveness of the techniques, as well as the management of the project are discussed in detail.

II. GROUP WORKING IN SPONSORED PROJECTS

Most of the sponsored software projects are constantly looking for ways to deliver better product in small amount of time. There are schedules to be worked out to achieve the goal of the project. Faced with limited information, limited time, the productivity of the software team becomes critical to project success [4]. However, dedicated group of people and systematic documentation in the development of the project, determine the outcome of the project.

It is evident that the real-life project assignments improve software development skills incredibly. Students need to experience the stress and adversity of working in a real-time project within a group [5]. Moreover, it will be a real-world experience when they are assigned to the roles and responsibilities given by the project leader.

People management, user interface design, configuration management, ethics, professionalism, requirements gathering, and project management are considered as the biggest gap areas between what is being taught in the school environment and what is required in the software industry [6]. However, the real-world software development in a group provides good opportunities for gathering requirements and designing the user interface. Also client based projects force students to increase their professionalism in their work life. Sponsored projects not only help the students, but also teach the group leader how to combine several people under single consolidated framework, how to manage students with different skills, ethics, and schedule.

III. DEVELOPMENT OF THE PROJECT AND PROJECT SCHEDULING

During the first week of the semester, all students were assigned to work on a project. Being a real-world software development project, it increases the motivation and commitment of the students. Due to security related issues, the clients were informed about the students. Furthermore, students were made aware of the confidentiality of the information that is related with the project.

Indeed, the project was started two weeks prior than the group held their first meeting. By the first group meeting, only the understanding and a diagrammatic representation of the project was delivered. In the first meeting, general idea of the project and the problem statement was explained to students and they were encouraged to get all the background of the

project and review previously used approaches. Also, they were notified that not only the students had to attend the class, but also had to join three meetings per week related with the project.

For the success of the project, and to understand the students' desire on the assignment, second meeting is crucial. Although the students are bound to be successful in the course, they can work ambitionless. Indeed funded project was extremely complex and without putting extra effort, productivity is just a fancy. In the second meeting, all students were prepared well, expressed their understanding of the project and made some comments on the existing document. Meanwhile, they were attending lecture sessions and following the Software and Software Engineering and Requirements Engineering topics. It is always much more beneficial to combine the theoretical parts of the system with practice. Therefore, because they learned the fundamentals of the requirements gathering from lecture, and had the understanding of the project, preparing Software Requirements Specification (SRS) document was assigned to each of student.

A. Requirements Gathering

The requirements document should cover all the functions of the product that needs to be performed; therefore, a well-prepared requirements document indicates all steps and the size of the project. Awareness of all the project scope and feedback from sponsored-client, it will be an ultimate ancillary to the project leader in the project scheduling. For example, if the requirements call for a large product, project leader suggest to dividing the job to phases and schedule oncoming events based on priorities.

Preparing a document individually always influence person to think by his own and force him to be creative while writing it. Moreover, combining individual work effort after going through separate documents lead to observe different approaches and it can propagate the scope of the project unthinkable paths and add some new functional requirements to the final SRS document.

After combining all requirements information in one common SRS document, it was a fact that the project itself was too much complicated, and there were a number of unknown pieces in it. Although a final deadline was given to deliver the final SRS document, the group didn't have a chance to produce it on time. Submission of the SRS document delayed couple of weeks than it was scheduled. The followings are the reasons for not being able to produce the complete SRS document in time.

- Missing information in delivered documents (from client)
- Comprehension of the complex project
- Owner's lack of technical background in some areas of the project
- Time constraints

Even after producing the SRS document, it was obvious

that the SRS was incomplete and there were number of issues that were needed to be addressed in the project. In client-sponsored projects, focus areas of the project can be changed constantly. During the analysis, design or even in implementation phase, one can discover new functional requirements that need priority than the others. Therefore, trying to obey the fundamental principle of having a “complete documentation” before advancing any further was very difficult to maintain. On the other hand, due to time constraints, waiting for the finalization of the SRS document to start the implementation is an open ticket for failure. Therefore, a quick assessment on the project planning and resource utilization with respect to the issues identified above, lead us to consider an agile development to obtain a prototype. Based on the group’s understanding of the project and feedback collected from the client, the first demo was built. Future enhancements are tolerated by avoiding extensive hard coding in the first prototype, since a degree of intelligence is planned to be introduced in the system.

B. Development Process

Due to complexity involved, the object-oriented approach was chosen for the implementation of the project. In order to specify, visualize, construct, and document the software architecture the Unified Modeling Language (UML) was used. The reason for using UML is to visualize the level of complexity involved and also help the students to understand what it takes to develop the project with some constraints. Basically use-case diagrams, class diagrams and sequence diagrams were provided for the project as part of the documentation. Use-case diagrams capture the behavior of a system, subsystem or class as it appears to an outside user [7]. Class diagrams show the classes of the system, their interrelationships (including inheritance, aggregation, and association), and the operations and attributes of the classes [8]. Class diagrams were used to visualize the detailed design model of the project. The sequence diagrams document the interactions between classes to achieve a result, such as a use case [8]. In the project, sequence diagrams helped to identify the relationships between components and the flow of logic within the system. ArgoUML [9] was used to build the diagrams.

Not only in the requirements phase, but also continuing to include the client in the development process helped ensure the developing product stays on target based on functional requirements. One way to do this is to quickly build a demo to present and get the feedback from client which is called rapid prototyping model [10]. The necessary documentation was delivered to client one week prior to visiting project group. By the meeting time, a prototype of the product framework and interface were designed and presented along with detail explanation of the algorithm including the assumptions to the client. This demo and presentation allow client to understand and experience the evolving look and feel of the product. The advantage of using customer feedback at the development stage can save significant amount of time required to perform

the necessary changes in the program later in the process. Closely working with client in all phases of the development introduces a better project management which produces an acceptable product.

In the development process, the effort was spent in three parts (Figure 1). As mentioned before, the project itself was highly complicated. Despite having documents and data about the current environment, the project was a research project and there were too many unknown issues to be addressed. Hence, 70% to 80% of the time was spent in group meetings, customer communication, requirements gathering, analysis, and design. Coding part covered the least time. After requirements gathering and design documentation, coding part didn’t require too much research on the technology to be used and coding to be carried out. Unit integration, regression and testing covered 10% to 15% of the time. Several test cases were developed and each case was tested with the tool and verified with the empirical study. In software engineering projects, not only the requirements documentation, but also the final report is very crucial for further analysis. Therefore most important parts of the project, such as algorithms, technology, both empirical and tool results on cases were documented in this stage to provide and maintain a reliable history.

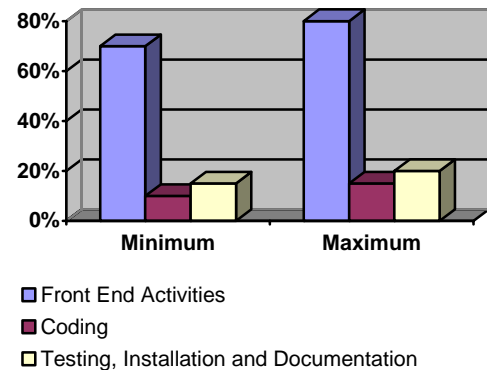


Fig. 1. Effort Allocation in the Project

C. Focus Group Sessions and People

The meeting sessions need to be carefully managed by group leader to make sure that all main contributions can be made during the allocated time [11]. The discussion can be a structured discussion, where the moderator dominates the meeting and only give his ideas and expecting to be accepted or it can involve with brainstorming techniques. In group sessions, the moderator acted as one member of the group and excited group for brainstorming. These meetings usually last one to two hours and have a predefined schedule. In this time of period, sufficient time was allocated for the participants to comprehend the issues and have a meaningful discussion and interaction.

In all group development software projects, it is important to be successful to have the right and motivated people working on it. Depending on the student’s background, participants may have less experience in the software

development area. It is important to take the time to find the best possible person, when assigning the staff to a particular job. Selecting of the correct candidate should not be limited only with their skills and knowledge, but also to candidates' work ethic and his contribution with the group are the other important things that need to be considered [4]. Since the project required a team work, group leader paid attention to the composition of each team in terms of developer's skills, task assignments, and compatibility. It often takes some time for group member to understand and deliver the task on time. Therefore, development concepts and UML language were taught before the implementation for the project.

Sometimes having the excellent staff, or making the best of existing staff is not enough to achieve the final point of the project, motivation is necessary to build the productivity. Not only the group leader, but also all students encouraged each other to finish the project on time and with no error.

Including group session meetings, internet technology was used often and this was one of the main reasons for decrease in number of meetings (Figure 2). All members of the group were part-time employees of the UALR and working in different departments. Due to individual schedule problems, bringing all group members together was not feasible every time. Therefore, sometimes rather than physical meetings, e-mail system was used for communication. Although too many aspects of the project were identified and discussed in electronic environment, sometimes it triggered some time loss, and confusion because of misunderstandings and the intensity of e-mails exchanged.

One other reason in decreasing the numbers of meetings, as the development of the project advances, was attributed to the familiarities gained in exhaustedly dealing with the problem. Commencing part of the projects needed more discussion and research. Once the competency level has reached by all members of the team, individual task assignment conducted independently in more efficient fashion in isolation. This helped to reduce the number of meetings scheduled frequently to discuss some development issues. Meetings organization has become delivery oriented at this point on.

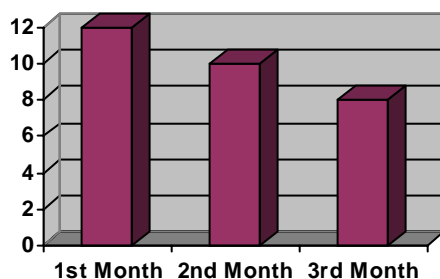


Fig. 2. Number of Meetings per Month

D. Project Scheduling and Tracking

Software development requires discipline and planning. Client-sponsored projects help students realize their responsibility, such as organizing themselves to work

together, taking the responsibility for the assigned job and deliver the product on time. An effective software process should define a collection of task sets designed to meet the needs of different types of projects [12].

Defining task sets is based on:

- Determining type of project
- Assessing the degree of rigor required
- Identifying adaptation criteria
- Computing task set selector value
- Selecting appropriate software engineering tasks

Feedback from client and group sessions automatically provided a set of tasks that need to done. Group leader made the task distribution to other groups members with deadlines based on their individual skills. Scheduling is useful to project leader prior to and during the project. A well prepared-schedule with requirements document points the finish time of the project months before it. Their ability to highlight the project's critical path and task slack time allows the project leader to focus more attention on the critical aspects of the project-time, costs and people in the group. However, due to constant changes in requirements specification and some unpredictable risks, those were not considered when the project commenced, led some delays in delivery time (Table 1).

TABLE 1
SOME PROJECT TASKS and THEIR SCHEDULE

Tasks	Planned Duration	Actual Duration
SRS Document	3 days	19 days
UML Use Case and Class Diagrams	2 days	On time
UML Sequence Diagrams	2 days	3 days
Information Gathering and Database Design	10 days	12 days
Coding	6 days	On time

IV. CONCLUSION

Looking at the performance of Master's degree students in their Software Engineering class, this course and the project seems to be a very positive impact on the students and project itself. This project was the first real world system development for most of the members in the group and it is important to ensure that this experience is more positive rather than negative. The project-based assignment played an important role in preparing students for job environment. It also improved the effectiveness of the Software Engineering course very much. Moreover, the product was much more functional than the client's expectation.

Learning from mistakes or failures is a strategy used in software engineering development. Lack of information to

understand and implement the product in early stages was significant in this project. When managing software development projects, awareness of the complexity of gathering requirements, preparing a well-written document, task distribution and scheduling can have a positive impact on productivity. In conclusion, project-based assignments help students to gather professionalism in their work and help them future real-world projects.

REFERENCES

- [1] Astin A. V., "Computer Development (SEAC and DYSEAC) at the National Bureau of Standards, Washington, DC," *National Bureau of Standards Circular 551*, U.S. Government Printing Office, Washington, DC, 1955
- [2] Villarreal, E.E. and Butler, D., "Giving Computer Science Students a Real-World Experience" *Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education*, ACM Press, Atlanta, Georgia, USA, 1998, pp 40-44
- [3] Polack-Wahl, J.A., "Incorporating the Client's Role in a Software Engineering Course" *Proceedings of the 30th SIGCSE Technical Symposium on Computer Science Education*, ACM Press, New Orleans, Louisiana, USA, 1999, pp 73-77
- [4] K. S. White, "Software Engineering Management for Productivity and Quality" *International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS) Proceedings*, Trieste, Italy, Oct. 1999, pp 317-319
- [5] Surendran, K., Young, F. H. "Teaching Software Engineering in a Practical Way", *13th Annual Conference of the national Advisory Committee on Computing Qualifications*, Te Papa, Wellington, New Zealand, 2000 NACCQ: 345-350
- [6] Lethbridge, T.C., "What Knowledge is Important to a Software Professional?" *IEEE Computer*, May 2000, Vol.33, Number 5, pp 44-50
- [7] UML-Class-Diagrams,
<<http://www.agilemodeling.com/artifacts/classDiagram.htm>>
- [8] Rumbaugh, J., Jacobsen, I., Booch, G., "The Unified Modeling Language Reference Manual", Addison-Wesley Object Technology Series, December 23, 1998
- [9] ArgoUML, <argouml.tigris.org>
- [10] Jurens, K. K., "Rapidprototyping's second decade" *Rapid Prototyping*, 1998, pp.1-3
- [11] Kontio, J., Lehtola, L., Bragge, J., "Using the Focus Group Method in Software Engineering: Obtaining Practitioner and User Experiences" *International Symposium on Empirical Software Engineering (ISESE)*, Aug 19-20, 2004, Redondo Beach, U.S.A, pp. 271-280
- [12] Maxim, B., R., "Project Scheduling and Tracking", Introduction to Software Engineering (CIS 375) Class Notes
<<http://www.engin.umd.umich.edu/CIS/course.des/cis375/ppt/lec5.ppt>>